DOCUMENT RESUME

ED 181 450 CS 205 071

AUTHOR Goldstein, Ira; And Others
TITLE Artificial Intelligence, Language and the Study of Knowledge.
INSTITUTION Massachusetts Inst. of Tech., Cambridge. Artificial Intelligence Lab.
SPONS AGENCY National Inst. of Education (DHEW), Washington, D.C.
PUB DATE Oct 75
GRANT NIE-P-75-0012
NOTE 82p.

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS *Artificial Intelligence; Communication (Thought Transfer); *Comprehension; Computer Assisted Instruction; *Educational Technology; *Language Processing; *Language Research; Teaching Techniques; Theories; Thought Processes

ABSTRACT

This paper studies the relationship of artificial intelligence (AI) to the study of language and the representation of the underlying knowledge that supports the comprehension process. It develops the view that intelligence is based on the ability to use large amounts of diverse kinds of knowledge in procedural ways, rather than on the possession of a few general and uniform principles. Specific topics covered in the seven chapters include AI as the study of intelligent processes; frames as a unifying approach to representing knowledge; the structure of an AI theory; self-knowledge; control structures; the Classical, Romantic, and Modern periods of AI; the impact of AI on education, and controversies about educability and innateness. (FL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED181450

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Ira Goldstein

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

Artificial Intelligence, Language and the Study of Knowledge

Ira Goldstein
Seymour Papert
Marvin Minsky

July 1975
Revised October 1975

This paper studies the relationship of Artificial Intelligence to the study of language and the representation of the underlying knowledge which supports the comprehension process. It develops the view that intelligence is based on the ability to use large amounts of diverse kinds of knowledge in procedural ways, rather than on the possession of a few general and uniform principles. We conclude with a discussion of how Artificial Intelligence may have a radical impact on education if the principles which it utilizes to explore the representation and use of knowledge are made available to the student to use in his own learning experiences.

The research and writing of this paper was sponsored under NIE Order No. NIE-P-75-0012. The views and conclusions contained are those of the authors and should not be interpreted as representing the policies, either expressed or implied, of the National Institute of Education or the U.S. Government.

Artificial Intelligence Lab.
Massachusetts Institute of Technology

CS205071

Contents

Chapter 1 -- Introduction

Artificial Intelligence and the Study of Language
Artificial Intelligence as the Study of Intelligent Processes
A Knowledge-Oriented rather than Power-Oriented Theory of Intelligence
Artificial Intelligence as a Procedural Theory of Knowledge
Educational Applications

Chapter 2 -- Frames as a Unifying Approach to Representing Knowledge

Case Grammars
Understanding a Sentence
The Procedural Idea of a Frame-Keeper
A Formal Realization of the Frame Idea
Frames for Representing Knowledge about Nouns
Extending the Frame Idea to a Theory of Context
Frames for Thematic Knowledge

Chapter 3 -- The Structure of an AI Theory

Issues in Computational Epistemology
Representing Declarative Knowledge
Representing Procedural Knowledge
Representing Knowledge in Highly Structured Packets
The Demon Solution to the Reference Problem
The Virtue of Particular Knowledge

Chapter 4 -- Self-Knowledge

Kinds of Self-Knowledge
The SHRDLU Program and the Blocks World
Answering Elementary Why Questions
Self reference in Grammatical Systems
Debugging Knowledge

Chapter 5 -- Control Structures

Controlling Grammatical Computations
Augmented Transition Networks
An Issue for the Future -- Deterministic Parsing

Chapter 6 -- The Classical, Romantic and Modern Periods of AI

A Change in Perspective
Unanswered Technical Issues in Frame Theory

Chapter 7 -- The Impact of Artificial Intelligence on Education

If the Machines Get Smarter, Will the Children Get Smarter?
Intelligent Computer Aided Instruction
The White Box Approach to Language Studies
The Articulate Learner
Epistemology and Education

Epilogue -- Controversies about Educability and Innateness

Bibiography

Introduction

Artificial Intelligence and the Study of Language

This essay is about the impact on the study of language of a new style of thinking about cognition known as *Artificial Intelligence* (AI). The goals of the essay are:

- to convey a sense of the intellectual personality of Artificial Intelligence; and

- to identify some trends in its development which we see as especially relevant to the study of language and of the role of an AI approach to language in education.

Artificial Intelligence as the Study of Intelligent Processes

Before we embark on the substance of this essay, it is worthwhile to clarify a potential source of confusion. For many, AI is identified as a narrowly focussed field directed towards the goal of programing computers in such a fashion that they acquire the appearance of intelligence. Thus it may seem paradoxical that researchers in the field have anything to say about the structure of human language or related issues in education. However, the above description is misleading. It correctly delineates the major *methodology* of the science, that is, the use of computers to build precise models of cognitive theories. But it mistakenly identifies this as the only purpose of the field. Although there is much practical good that can come of more intelligent machines, the fundamental theoretical goal of the discipline is understanding intelligent processes *independent* of

their particular physical realization.

Naturally, understanding intelligence abstractly has intimate interactions with understanding it in its natural form. Thus, AI and psychology have close ties. Indeed, AI has been called "theoretical psychology" [Newell 1973]. AI draws from psychology a set of basic concerns -- understanding language, perception, memory, problem solving. Psychology, in turn, acquires a new framework of computational ideas for expressing cognitive theories. Indeed, a major purpose of psychology becomes the discovery not of the class of programs that could possibly explain a given intelligent behavior (this is the archetypical concern of AI), but rather the particular program that a given individual actually possesses.

A Knowledge-Oriented rather than Power-Oriented Theory of Intelligence

Given the general perception of AI as the computational study of intelligence, it still remains to identify the particular set of questions and theoretical constructs that a researcher in the field brings to bear on cognitive problems. In the earliest years of the field (i.e. during the late fifties and early sixties), the hope was that a few simple but very powerful techniques could be identified that together could be used to create intelligent programs. During this period, theorem proving and search played a primary role. Indeed, to many AI has become identified as the science of *heuristic search*, that is, the study of procedural techniques for exploring state spaces too large to be explored in an exhaustive fashion. During these early years, an AI paper on language might have begun by representing the state space by means of a formal grammar and then have focussed on the issue of achieving successful parsings in reasonable times. [Green 1963; Lindsay 1963].

Today there has been a shift in paradigm. The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent *large amounts of knowledge in a fashion that permits their effective use and interaction.* This shift is based on a decade of experience with programs that relied on uniform search or logistic techniques that proved to be hopelessly inefficient when faced with complex problems embedded in large knowledge spaces. The solution from the current point of view is for the problem solver (whether man or machine) to know explicitly how to use its knowledge -- with general techniques supplemented by domain specific pragmatic knowhow. Thus, we see AI as having shifted from a *power-based* strategy for achieving intelligence to a *knowledge-based* approach. Minsky and Papert characterize these two points of view as follows:

> The POWER strategy seeks a generalized increase in computational power. It may look toward new kinds of computers ("parallel" or "fuzzy" or "associative" or whatever) or it may look toward extensions of deductive generality, or information retrieval, or search algorithms - things like better "resolution" methods, better methods for exploring trees and nets, hash-coded triplets, etc. In each case the improvement sought is intended to be "uniform" -- independent of the particular data base.
>
> The KNOWLEDGE strategy sees progress as coming from better ways to express, recognize, and use diverse and particular forms of knowledge. This theory sees the problem as epistemological rather than as a matter of computational power or mathematical generality. It supposes, for example, that when a scientist solves a new problem, he engages a highly organized structure of especially appropriate facts, models, analogies, planning mechanisms, self-discipline procedures, etc. To be sure, he also engages "general" problem-solving schemata but it is by no means obvious that very smart people are that way directly because of the superior power of their general methods -- as compared with average people. Indirectly, perhaps, but that is another matter: a very intelligent person might be that way because of specific local features of his knowledge-organizing knowledge rather than because of global qualities of his "thinking" which, except for the effects of his self-applied knowledge, might be little different from a child's.
>
> [Minsky-Papert 1974 p. 59]

It is worthwhile to observe here that the goals of a knowledge-based approach to AI are

closely akin to those which motivated Piaget to call his research center "Centre d'Epistemologie Genetique." Or more precisely, to his reason for calling himself an "epistemologist" rather than a psychologist. The common theme is the view that the process of intelligence is determined by the knowledge held by the subject. The deep and primary questions are to understand the operations and data structures involved. The physical (biological or electrical) mechanisms underlying them are not seen by us or by Piaget as the source of intelligence in any structural sense (although intelligence could not happen without some such embodiment.)

Perhaps the knowledge paradigm is most clearly characterized by a developmental perspective. Naturally, intelligence is impossible without a certain basic "hardware" capability. But our position and Piaget's is that this capability is already present in the very young child. The subsequent development of intelligence is not due to the emergence of new biological capabilities in the individual. Rather, increased performance is due to the acquisition of knowledge. This knowledge can range from specific facts about a domain to general problem solving schemata. Indeed, it may even be knowledge about how to learn (for example, facts about debugging techniques). Some facts result in only local improvements in the individual's capabilities, while others may have global effects, giving rise to a stage transition in Piagetian terms. From the computer standpoint, we see the need for some improvements in speed and memory capabilities, especially for such special purpose applications as vision and speech. But the fundamental difficulties facing researchers in the field today are not limitations due to hardware, but rather questions about how to represent large amounts of knowledge in ways that still allow the effective use of individual facts.

Artificial Intelligence as a Procedural Theory of Knowledge

The reader may perhaps wonder at this point how AI is to have an identity of its own if the knowledge strategy is pursued, i.e. if it asks only for explicit representations of the knowledge of particular domains. Isn't this simply the concern of the specialized field of study associated with that domain? How does this notion of AI, for example, distinguish it from linguistics when the domain is that of language competence?

The answer lies in identifying two concerns of AI which are uniquely *procedural* in nature. The first is that the knowledge of a particular domain must be represented in a form that is useable. This is what is meant by *effective* in the above definition. A linguist may content himself with the specificiation of some attribute of grammatical knowledge by defining the appropriate transformation: an AI researcher must go one step further by asking how this transformation can be woven into the total web of linguistic information in such a way that the comprehension or generation process is possible in reasonable computational times. Thus, AI brings to the traditional study of a given field the quest for identifying *pragmatic* knowledge, namely knowledge of "how" traditional facts are to be used.

There is a second concern which distinguishes AI from traditional studies of a given domain. This is its concern for identifying the manner in which competence in different domains interact. Again drawing upon an example from the study of language, a grammarian can legitmately focus on attributes of grammar divorced from the semantics or use of language. The essential concern of AI, however, is to construct frameworks in which all of these diverse kinds of knowledge can successfully interact in order to allow the complete comprehension process to take place. Thus, both

concerns which make AI unique arise from its responsibility to structure knowledge into procedural systems that can actually solve problems.

Organization of this Paper

Our goal in the remainder of this paper will be to provide the reader with insight into the structure of theories currently being formulated in AI within this knowledge-based paradigm. Chapters 2 and 3 illustrate the AI approach by discussing a particular example -- frame theory -- which many people in the field are currently exploring and find quite exciting. Chapter 2 introduces the subject by approaching frame theory as an AI extension of the traditional linguistic notion of case grammars from a narrowly syntactic theory to a general representation for knowledge. Chapter 3 discusses the various attributes of AI theories that frames exemplify. In chapters 4 and 5, we deepen our view of AI as a computational style of epistemology by discussing two aspects of cognition and language to which AI has made unique contributions -- the role of self-knowledge and the importance of control structure. Chapter 6 concludes our theoretical discussion by briefly viewing in perspective the intellectual history of the field.

These topics do not by any means exhaust the contribution of AI to the study of language. But we believe they provide an excellent illustration of the AI methodology. And that seems to us the only serious purpose to be served by a paper of this length that is accessable both to students of the field and to non-specialists.

Throughout the paper, we avoid extensive excursions into formalism and jargon. More

seriously, we also ignore, or pass over in a cursory way, a variety of serious technical issues. We do this in order to focus attention on the overall enterprise of a computational investigation of knowledge. We feel especially justified because of the tendency in the literature to focus on narrow techical issues. Nevertheless, to compensate for this broad focus, we provide in chapter 6 a list of the technical questions which we feel are critical for the reader who wishes to pursue this point of view more deeply. Also, the reader should recognize that the various research we point to as illustrating our position have many important differences. However, we feel that the critical need at this time is to develop a feeling for the paradigm which characterizes AI as a method for investigating knowledge and language, and it is to this goal that we direct our efforts. Finally, we and others in the AI community expect to develop further the ideas raised in this paper. The interested reader is invited to write us for references to the latest research in this area.

Educational Applications

The final part of the essay (chapter 7) will touch schematically on some of the consequences for education of Artificial Intelligence research on natural language. These consequences are of very different sorts. Some of them, no doubt for most people the most obvious, are consequences of actually possessing machines capable of understanding what a student says. Obviously such machines would greatly improve the quality of "Computer Aided Instruction" (in the sense of tutorial programs, programmed instruction administered by computers, "drill and practice", computerized advice, etc.)

Much deeper consequences, however, are related to the intellectual content of Artificial

Intelligence research on natural language. As a new kind of cognitive theory it has obvious implications for the education. It is our personal opinion that these consequences will very deeply transform the theory and practice of education. Although we will not try to argue for this position at length, we shall try to convey the flavor of the perceptions on which it is based.

Frames as a Unifying Approach to Representing Knowledge

Case Grammars

Our goal in this chapter will be to illustrate how the powerful linguistic notion of case grammars has been generalized by AI to support a theory of knowledge that allows the representation of procedural knowledge for how to construct a parse as well as the expression of related semantic, contextual and thematic facts.

> In fact the frame theory which we shall discuss draws upon more than purely linguistic considerations. In the original formulation by Minsky [1975], the notions of frames and frame systems were developed in equal measure from problems arising from the construction of adequate theories for vision, for memory, for logical reasoning and for the comprehension of language. However, we shall narrow our focus in this chapter to consider the utility of frames solely in relation to linguistic knowledge. In the next chapter, we step back from the realm of language to consider a variety of larger issues that characterize the AI approach to representing knowledge.

The case frame approach to linguistics as developed by Fillmore [1968] and others constrains the enquiry to the realm of syntactic knowledge. Within this sphere, the basic idea is to organize structural knowledge of the sentence around the verb. The result is a *case grammar* which specifies for each verb the cases which it requires. As an illustration, the following phrase structure grammar specifies the underlying case structures developed by Celce-Murcia [1972] and utilized by Simmons [1973] in his language comprehension and generation programs.

S => MODALITY + PROPOSITION
MODALITY => TENSE, ASPECT, FORM, MOOD, MODAL MANNER, TIME
PROPOSITION => VERB + (CASEARGUMENT)0
VERB => run, walk, break, etc.
CASEARGUMENT => CASERELATION + [NP|S]
NP => (PREP) + (DET) + (ADJ)0 + (N) + N + (S|NP)
CASERELATION => casualactant, theme, locus, source, goal

[Simmons 1973 p. 68]

The general syntactic form for the case structure of a sentence is given in figure 1 .

In order to illustrate this, let us use the traditional linguistic device of choosing a representative sentence around which to organize our observations. Hence, S1:

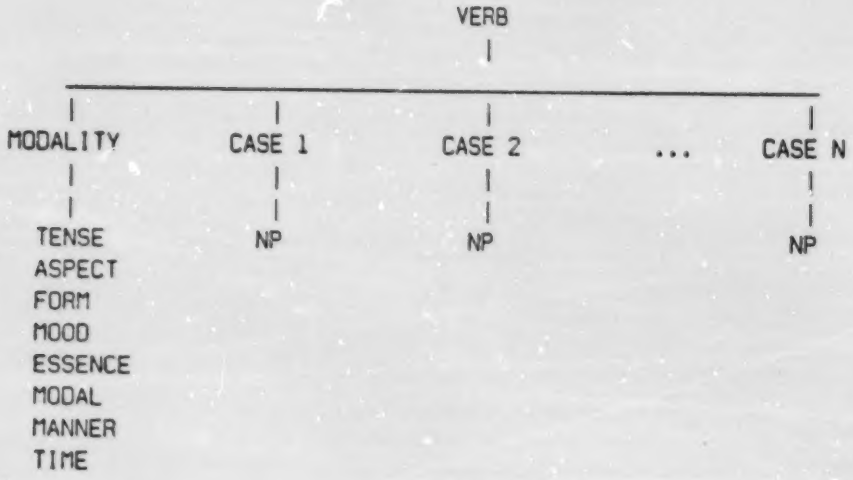
(S1) John tickled the girl with the feather.

For S1, a traditional phrase-structured grammar might produce the parse illustrated by figure 2 while a case-oriented approach would represent the sentence as in figure 3 . Figure 3A shows the case relations for this verb following Celce-Murcia while figure 3B follows the formulation of [Chafe 1970].

Understanding a Sentence

To make the linguistic enterprise tractable, the linguist deliberately ignores the issue of understanding an utterance beyond the narrow concern of identifying its syntactic structure. But AI is forced to address this larger issue if computational models are to be designed that actually act appropriately in response to the utterance. Thus, AI is willing to consider the senses of understanding S1 illustrated by the following three listeners:

> A five year old child, who has a clear image of John using the feather to evoke giggles.



The Cases with their Intuitive Meanings:

CAUSALACTANT1:	Actor, Manipulator, Causal Agent
THEME:	Object
CAUSALACTANT2:	Instrument, Second Causal Agent
LOCUS:	Location or Path of Activity
SOURCE:	Origin
GOAL:	Destination

The Modalities:

TENSE:	Present, Past, Future
ASPECT:	Perfect, Imperfect
FORM:	Simple, Emphatic, Progressive
MOOD:	Declarative, Interrogative, Imperative
ESSENCE:	Positive, Negative, Indeterminate
MODAL:	May, Can, Must
MANNER:	Adverbial
TIME:	Adverbial

Figure 1 -- Case Structure following Celce-Murcia from [Simmons 1973]

(S1) JOHN TICKLED THE GIRL WITH A FEATHER.

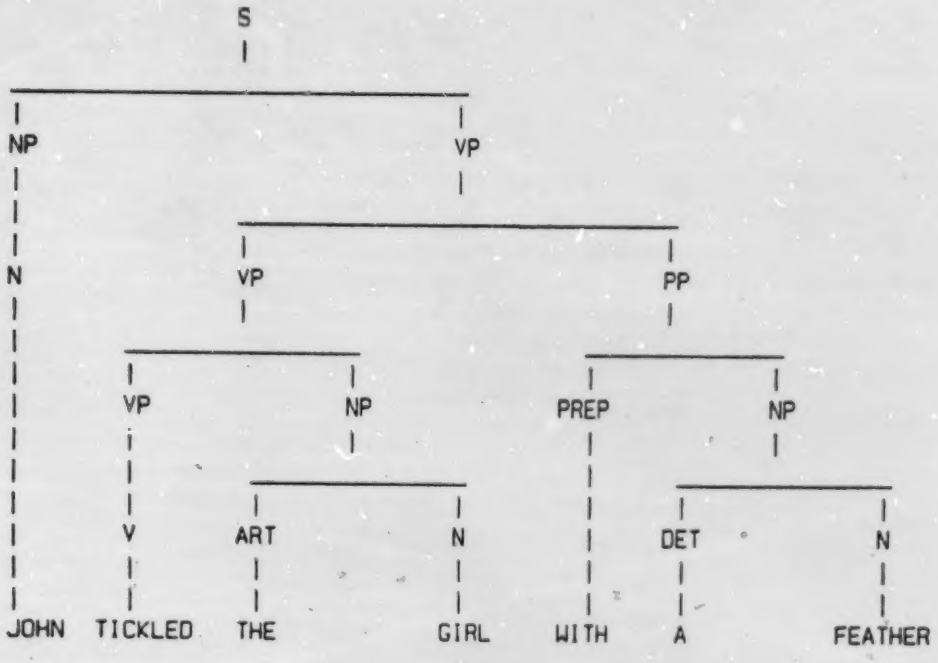


Figure 2 -- A Phrase-Structured Parse of S1

```
TICKLE:        CAUSALACTANT1 John
               THEME         the girl
               CAUSALACTANT2 a feather.
```

Figure 3 A -- A Case-Structured Parse of S1 following Celce-Murcia

```
TICKLE:        AGENT         John
               PATIENT       the girl
               INSTRUMENT    a feather.
```

Figure 3 B -- A Case-Structured Parse of S1 following Chafe

A slightly more sophisticated listener capable of taking account of a context such as: "All the girls in the room had strange hats. One was decorated with a large feather, another with a towering confection of artificial fruit. *John tickled the girl with the feather.*"

A linguist who understands the sentence as an example of a well known ambiguity in parsing.

It is obvious that all the subjects use some grammatical knowledge. For example the order of the words tells them that it was John who did the tickling and not the girl. But grammatical knowledge alone is not sufficient even to account for the syntactic parsing of the sentence, i.e. the interpretation of the phrase "with the feather" as adjectival or adverbial. This point is made clear by contrasting the sentence with S2:

(S2) John tickled the girl with the funny nose.

Even our five year old would not be likely to read this as if John used the nose as a tickling instrument. The difference is not puzzling to common sense: after all the child knows that feathers are good for tickling while noses are more likely to be used for smelling or as personal identifiers. But the difference should trouble the cognitive theorist who knows how to describe syntactic structures in fine detail but has no better description than naive common sense of the child's knowledge about feathers and noses. And the difference is relentlessly present for the computer-oriented theorist who absolutely <u>needs</u> to find ways to formalize it.

The Procedural Idea of a Frame-Keeper

Let us begin our generalization of case grammars by discussing informally how a procedural case theory might be organized. In the next section, we will discuss work by Riesbeck [1974] that represents one formal realization of this point of view.

In our informal mini-theory there is associated with each important word in the sentence an active agent which is strictly defined as a computational procedure but which we might just as well anthropomorphize (for the moment anyway) as a person. Let's call them *Frame-Keepers* for reasons which will soon become apparent. The process of understanding the sentence will consist of an interaction between these Frame-Keepers wherein each tries to fulfill certain needs. For example the needs of the TICKLE-FRAME keeper include:

Find the agent (the person responsible for the tickling)!

Find the patient (the person being tickled)!

Find the instrument (the means by which the agent is tickling the patient)!

So the TICKLE-FRAME keeper is seen as actively seeking to fill these slots written into his definition. If all of them can do so to their satisfaction (an idea which we will pursue shortly) the sentence is to be considered as parsed.

The essence of the AI extension of the case frame idea is that the frame for the verb contain not only syntactic information about the expected location of the underlying cases, e.g. in a declarative sentence that the subject is the agent, the object the patient and the "with" prepositional

phrase the instrument, but also semantic and pragmatic facts. Thus, we imagine that for our tickle-frame, the frame-keeper has access to world knowledge that states that feathers are typical tickling instruments, while other kinds of objects, for example, dumbbells, are not. Hence, it would be the *tickle-frame keeper* that would be in a position to reject for S2 that the "with" phrase refers to the tickle instrument because it knows that one generally does not tickle with noses.

In the next section, we discuss an existing program that associates both syntactic and semantic knowledge with individual words in order to comprehend the sentence. Following that, we indicate how the frame idea can be extended to nouns. Given that extension, the process of comprehending S2 becomes clearer as not only does the tickle frame reject the nose phrase, but the girl frame accepts it as a legitimate descriptor of people. Thus, the system has two sources of evidence both supporting one another regarding the intended parse.

A Formal Realization of the Frame Idea

Our discussion of the TICKLE-FRAME and frame-keepers represents the kind of concern for active knowledge that AI brings to linguistic studies. C. Riesbeck [1974, 1975] has designed and implemented a program that, although not frame-based, shares many important properties with the approach outlined above. Riesbeck's parser maps natural language into the Conceptual Dependency structures designed by Schank [1973]. S3 is typical of the sentences that Riesbeck's system analyzes.

(S3) John gave Mary a beating

Procedural information is associated with each word in the sentence, which is triggered when the

parser encounters the word in its left-to-right analysis. This information constructs an instantiated case-frame-like structure. It also contains expectations that serve to disambiguate multiple word senses and properly decode the syntactic cues that indicate the fillers for each slot in the case-frame associated with the verb. A trace of the analysis of the parser is given in figure 4 .

The system, called MARGIE, of which Riesbeck's parser was a part, exhibited its understanding by generating paraphrases of the original sentence and by making deductions based on it [Schank 1972]. The basic operation was to comprehend the sentence by mapping it into the conceptual dependency representation developed by Schank [1973]. Figure 5 illustrates the conceptual dependency diagram generated for the above sentence. The details of Schank's representation are not important here, except to note the manner in which the system has understood that the sentence refers to a series of actions by John, involving the motion of his hand. Riesbeck's parser has not erroneously interpreted "give" to mean that John is transfering the possession of something to Mary.

Frames for Representing Knowledge about Nouns

The case frame idea in its original linguistic form was oriented towards representing knowledge about verbs. AI pursuing the goal of comprehensive theories of knowledge has generalized the idea to allow frames for nouns.

Consider again the sentence "John tickled the girl with the feather." We imagine that part of the knowledge for properly understanding this sentence is that the verb-frame for tickle has as a

STEP	WORD READ	REQUESTS WAITING	REQUESTS TRIGGERED	ACTIONS TAKEN
Ø	none	1-is there an NP?	none	none
1	John	1-is there an NP?	1	assume "John" is subject of the verb to follow
2	gave	2-is the current NP a human? 3-is the current NP an object? 4-is the current NP an action? 5-true	5	assume the word "to", if it appears, introduces the recipient of the "giving"
3	Mary	2-is the current NP a human? 3-is the current NP an object? 4-is the current NP an action?	2	assume Mary is the recipient of the "giving"
4	a	3-is the current NP an object? 4-is the current NP an action? 6-true	6	save the current list of requests and replace it with: 7-does the current word end an NP?
5	beating	7-does the current word end an NP?	none	none
6	period	7-does the current word end an NP?	7	build the NP "a beating" and reset the list of requests
7	none	3-is the current NP an object? 4-is the current NP an action?	4	assume the NP action is the main action of the clause, the subject (John) is the actor and the recipient (Mary) is the object

Trace of the Riesbeck Parser on "John gave Mary a beating."
From [Riesbeck 1974 p. 31]
Figure 4

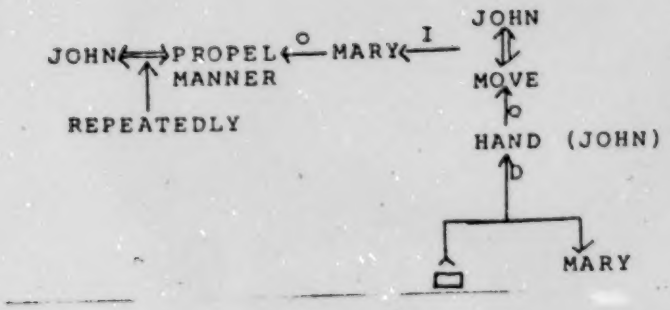


Figure 5

Conceptual Dependency Representation of the meaning of the sentence:

"John gave Mary a beating."

From [Riesbeck 1974, p. 108].

typical instrument "feathers." Such verb-centered knowledge is supplemented by frames for such nouns as "girl" which contain pointers to sub-frames which describe typical appearances, purposes, activities, etc. The appearance sub-frames would indicate that girls do not "have" feathers. They might under closer scrutiny indicate that "feathers" are a possible but unlikely kind of dress. However, the strong default for "feathers" as a typical tickling instrument would predominate.

> In the previous paragraph, we wrote as if the tickle frame could have *direct, explicit* knowledge about feathers. A fundamental and pervasive class of problems concerns the relative feasibility of achieving this as compared to various schemes for *indirect* reference to feathers via some *super-set* or *description*. For example, the tickle frame might know that "long, soft objects" are good instruments for tickling and somehow use this as a means of finding a connection to feathers. Most early workers in AI (like most contemporary psychologists) were strongly averse to the direct approach and so a great deal of attention has been given to the invention of "search" or "information retrieval" methods to mediate the indirect approach. Current thinking (especially amongst frame-theorists) leans towards very much more direct knowledge than was previously considered feasible. This places a heavy demand on *memory*, and less on retrieval mechanisms. This appears to some to be "heavy-handed". But we do know that human memory has a very large capacity when appropriately used and no one has been able to propose very sophisticated retrieval mechanisms. So perhaps it is nature (rather than the theorists) who has adopted the "heavy handed" solution.

A related example shows where noun-frame information might predominate. Consider the three sentences:

(1) I frightened the dog with the loud noise.

(2) I frightened the dog with the brown fur.

(3) I frightened the dog with the loud bark.

We believe most readers would understand (1) by assuming that the "loud noise" was the instrument of "frighten". This would be because of both the verb-frame knowledge that loud noises often can be used to frighten and the noun-frame knowledge associated with dogs that

indicate that "noise" is not "possessed" by them. On the other hand, in sentence (2), the noun-frame knowledge for dogs would surely indicate that they typically have "fur", hence the phrase "with the brown fur" would be understood to be modifying the noun "dog". In sentence (3), we suspect that our population of readers would be divided, depending on whether their frame for "frightening" contained the typical instrument of emitting a loud noise, describable as a bark, with a stronger emphasis than their frame for "dogs" contained the default property of emitting noise by barking. (This extension of frames to nouns has not yet reached the point of development in the form of programs. However, a recent thesis by C. Drake [1975] extends the details of the theory beyond the outline provided by Minsky to a form that is closer to the implementation state.)

Extending the Frame Idea to a Theory of Context

Procedural frames which are not restricted to verbs are powerful tools for dealing with the set of problems traditionally called "context". Indeed, once we break away from the verb-centered sentence as the unit for analysis, it becomes possible to talk about "scope" in a possibly very large text.

This too is a topic of current research and there are few programs implemented as yet. One of the systems that does exist, however, is Schank's successor to the MARGIE system that generates paraphrases of simple stories [Schank 1975]. The essential theoretical idea in the new system is the existence of frames (called scripts by Schank) that describe typical activities in terms of his Conceptual Dependency primitives. An example is the following informal sketch of a frame for going to a restaurant:

```
Script: restaurant
Role: customer, waitress, chef, cashier
Reason: to get food so as to go down in hunger and up in pleasure.

Scene 1 Entering
        PTRANS - go into restaurant
        MBUILD - find table
        PTRANS - go to table
        MOVE   - sit down

Scene 2 Ordering
        ATRANS - receive menu
        ATTEND - look at it
        MBUILD - decide on order
        MTRANS - tell order to waitress

Scene 3 Eating
        ATRANS - receive food
        INGEST - eat food

Scene 4 Exiting
        MTRANS - ask for check
        ATRANS - give tip to waitress
        PTRANS - go to cashier
        ATRANS - give money to cashier
        PTRANS - go out of restaurant
```

[Schank, 1975, p. 117]

The image of understanding a story is mapping the sentences of the story into the actions described in the frame. Unstated facts described in the frame are assumed. Thus, mapping a story in which the sentence appeared: "John left a tip" into this frame would allow the system to know that the "tip" was left for the waitress, a fact contained in the frame but not in the story.

Frames for Thematic Knowledge

Our previous examples have discussed extensions of the case frame idea to incorporate into the frame a larger fraction of the knowledge associated with comprehending a sentence than is included within the purely grammatical realm. We would like to conclude this chapter on frames as a theory of knowledge by extending our representation beyond the sentential level to include the kinds of knowledge that link sentences together. This represents the frontier of current research and so we will have few formal examples and have to content ourselves with some observations which indicate the direction in which we expect this research to go.

To develop this notion of thematic frames, let us consider the problem of understanding a fairy tale. Our goal is to represent the knowledge that links sentences in a story together into a coherent whole. To begin, we will first examine some work by Rumelhart [1975] which addresses this question, but utilizes a phrase-structured grammar (augmented by semantic interpretation rules) to represent the structure of the story. Typical of the rules that his system possesses are:

RULE 1: STORY -> SETTING + EPISODE.

The setting is a statement of the time and place of the story as well as an introduction for the main characters, e.g. "Once upon a time, in a far away land, there lived a good king."

RULE 2: SETTING -> (STATIVES)*

Episodes consist of events followed by reactions:

RULE 3: EPISODE -> EVENT + REACTION

An event is the most general category of the grammar:

RULE 4: EVENT -> EPISODE | CHANGE-OF-STATE | ACTION
| EVENT+EVENT

A reaction consists of an internal and an overt response:

RULE 5: REACTION -> INTERNAL-RESPONSE + OVERT-RESPONSE.

There are a large number of internal-responses. Two are:

RULE 6: INTERNAL-RESPONSE -> EMOTION | DESIRE.

Overt-responses can be either an action or a series of attempts:

RULE 7: OVERT-RESPONSE -> ACTION | (ATTEMPT)*

Attempts also have an internal structure:

RULE 8: ATTEMPT -> PLAN + APPLICATION.

The plan is developed to accomplish a desire. The application is the actual execution of the plan.

For each of these rules, there are semantic interpretation rules that indicate for example that rule 1 means that both the setting and the episode must be present while rule 3 (with the same syntactic "+" connective) indicates that the event is the cause of the reaction and precedes the reaction in time.

Let us now ask how the skeleton represented by this story grammar might be extended into a frame system. The basic change is the addition of large amounts of highly structured and particular knowledge. This information which we shall call *default knowledge* includes information about the various aspects of the story, e.g. setting, episodes, internal and external responses. Indeed, the proposal is that the system knows a few "favorite" stories in extensive detail. Thus, with only a brief introduction to the story such as "Once upon a time and long ago", the listener is able to bring to mind an elaborate setting corresponding to his primary default -- perhaps the Camelot setting, the Black Forest setting, or the humble cottage depending on the idiosyncracies of the frame system of the individual listener. Of course, conflicts will arise from

difference between the story and the default, causing one of two results: either exceptions are entered into the evolving database representing the system's comprehension of the story, or, if the conflict is too radical, the entire default setting is replaced with another. In either case, the result is that far more detail than is actually stated in the story is assumed by the listener through this mechanism of default knowledge.

Just as there can be default knowledge about states such as setting, so too can there be default knowledge about the expected sequence of episodes in the story, where each episode is itself a frame (e.g. a frame for the initial state -- a damsel in distress, followed by a sequence of frames describing a typical rescue operation by a handsome prince and hindered by a wicked witch). This organization of frames into state action sequences is essentially the Script idea of Schank's discussed earlier.

Default knowledge about expected themes, morals, narrative structures may also be included. We expect, for example, for a fable to have a moral, for the prince and princess to live happily ever after in a children's fairytale (though not in the original primitive folklore).

The point of this discussion is to articulate this model of comprehension in which the process is not one of literally understanding the text, but instead is more one in which the text triggers rich, highly structured knowledge packets that supplement the literal content, provide expectations for the remainder of the story and place the story in a context of related knowledge. Understanding is seen as essentially a process of evoking and then debugging existing knowledge packets.

The Structure of an AI Theory

Issues in Computational Epistemology

At this point, we have seen how the linguistic theory of case grammars has been extended by AI to become a knowledge-oriented theory of language. In this chapter, we step back from a purely linguistic orientation in order to gain a perspective on the general theory of intelligence and knowledge embedded in this point of view. Specifically, we shall examine:

(1) the manner in which frames exhibit a readiness to use new, flexible and complex data types;

(2) the fashion in which frames represent a synthesis of procedural and declarative knowledge;

(3) the sense in which the use of an anthropomorphic metaphor has become a technical device in AI theory;

(4) the trend towards the representation of knowledge in larger, more structured units (which frames exemplify);

(5) a concern for allowing interactions between diverse kinds of knowledge;

(6) a desire for expertly controlled processes;

Representing Declarative Knowledge

A central concern of AI is the construction of new datatypes and the exploration of their capabilities. The frame idea is still evolving in AI, and so it is difficult to give a precise definition. Minsky gives the following description of a frame in his original paper on the subject:

> We can think of a frame as a network of nodes and relations. The "top

> levels" of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many terminals -- "slots" that must be filled by specific instances or data. Each terminal can specify conditions its assignments must meet. (The assignments themselves are usually smaller "sub-frames.") Simple conditions are specified by markers that might require a terminal assignment to be a person, an object of sufficient value, or a pointer to a sub-frame of a certain type. More complex conditions can specify relations among the things assigned to several terminals.
>
> [Minsky 1975, p. 212]

We see from the above description that the traditional *declarative* data structures of objects, properties and relations plays an important role in the definition of a frame. This aspect of frame theory has its roots in earlier work in AI on the representation of knowledge as attribute-value pairs (e.g. the BASEBALL program [Green 1963]) and as semantic nets [Quillian 1968]. The notion of a Semantic Net has already had an important impact on cognitive psychology, c.f. work by Anderson and Bower [1973] and Norman [1975]. The additional ideas added by frame theory which generalize the notion of a Semantic Net are:

> (1) the organization of these relations, not into a uniform net, but into a highly structured set of contexts (the frames themselves);
>
> (2) the inclusion of defaults for properties and relations representing the expected values and supplying common sense knowledge about the context (possibly not supplied by the actual text); and

(3) the infusion of procedural knowledge which we shall discuss below.

Representing Procedural Knowledge

In the preceding chapter, we saw informally how procedural expertise is part of a frame. This expertise took the form of a *Frame-Keeper*, a procedure whose action is to examine the current context for data that can fill the slots of the frame. To further clarify the concept, we begin with a gedanken example of how an AI program might emulate a certain aspect of the way a mathematician thinks. The example goes right into the domain of the logician and proposes a very different way to cope with the existence of paradoxes such as the Burali-Forte paradox in mathematics and such paradoxes in ordinary language as the use of self-reflexive assertions such as "All Cretans are liars" (spoken by a Cretan and meaning: Cretans never tell the truth). The traditional logician's recourse is to seek an axiomatization from which no such contradictions can be deduced. Our theory of what an working mathematician actually does is to use principles of reasoning which could easily lead to paradoxes if they were to be applied in that direction. But the mathematician knows this and refrains from using them in directions known to be dangerous. It is as if he posted a sentry near dangerous places in his intellectual terrain. The job of the sentry is to issue warnings if any reasoning process comes too close to the danger spot. In AI we see as our job developing theories in which the concept of such a sentry can be easily formalized.

The anthropomorphic idea of posting a sentry is typical, we maintain, of the way in which sensible people have thought intuitively about this kind of problem. The contribution of computer

science is to show how to translate this intuitive matter into a technical and well defined form. If one represents a knowledge system as a set of propositions, the idea of introducing the sentry is outrageous, or at best a literary conceit. But if the knowledge system is represented in the first place as a set of interactive processes, the sentry is merely one more process. Naturally one has to be sure that this new proces. is well defined. But that is a technical problem. The conceptual change is that it is now perfectly natural and coherent to introduce such entities.

> From this perspective, a fundamental contribution of AI to epistemology is made clear; and that is, the systematic introduction of active agents into epistemological theory constructions, so that, for example, an item of knowledge, a concept, or whatever, is seen as an active agent rather than a passive manipulatable object.

The gedanken example from mathematics seemed to us well suited to illustrate the force of this concept in an intuitive way. But there are real applications applied to quite real problems. A well known example is Charniak's method for dealing with reference problems in children's stories. To illustrate this, consider the following story fragment from [Charniak 1972].

> Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a kite. "Don't do that," said Penny. "Jack has a kite. He will make you take *it* back."

The goal is to construct a theory that explains how the reader understands that "*it*" refers to the new kite, not the one Jack already owns. Purely syntactic criteria (such as assigning the referent of "*it*" to the last mentioned noun) are clearly inadequate, as the result would be to mistakenly understand the last sentence of the story as meaning that Jack will make Janet take back the kite *he already owns*.

Resolving reference is a traditional problem of linguistics that cannot be handled by purely syntactic techniques. Our solution will involve the realization of our intuitive "frame-keeper" as a precise procedural entity, the *demon*. This is a production-like entity that is activated by some aspect of the state of the computational world within which it is defined. But we will postpone the details of this solution until the next section in order to first give some perspective on the underlying AI philosophy.

Representing Knowledge in Highly Structured Packets

The fundamental frame assumption is the thesis that *fewer situations than we think are really as new as we think*. Most situations in which people find themselves have sufficient in common with previously encountered situations for the salient features to be pre-analyzed and stored in a situation-specific form. Whenever this happens we are able to use "pre-digested", highly particular knowledge -- an explanation in sharp contrast with the views of intelligence propounded by theorists who would prefer to believe that our minds work (and have to work!) in a deductive fashion from very general principles.

For our *kite* example, it is clear that one cannot know that *"it"* refers to the new kite without knowledge about the trading habits of our society. One could imagine a different world in which newly bought objects are never returned to the store, but old ones are. The question we raise here is how this knowledge might be represented, stored and made available to the process of understanding Charniak's story. To see the spectrum of possible theories consider two extremes:

An Extreme in the Frame Direction:

The listener has stored away in long term memory a frame for birthday parties in which certain events and problems are explicitly represented. These include the problem of choosing a present with quite explicit reference to the three or four "possible dangers to be thought about" including "recipient won't like it and will be disappointed" and "recipient already has one and will return it".

An Extreme in the Anti-Frame Direction

From a strictly logical point of view it is not necessary to have a pre-constructed party frame in order to assign the proper referent to "*it*". It is unnecessary to have been to a party or have given a present. It would be sufficient to have stored away for general use items of information such as:

(1) ((x has been bought at store y) and (x is in good condition))
implies (store y will exchange x for z)

(2) ((x is new) and (x has not been used))
implies (x is in good condition).

The advantages of this method (if it is workable) are apparent: less memory space, easier updating, modularity, etc. The major disadvantage is that we then have to face the question of

how these two items of information are chosen from the presumably very large stock of knowledge, presumably including many other items rather similar to these. How does the interpretive system know which items to use?

An even more serious problem takes the form: how does the system know that any appeal to knowledge about the world is necessary? Are we to assume that simple syntactic rules (e.g. based on word order and proximity) are never used? But if they are sometimes used, what restrains them in this case?

By providing possible answers to this question, the frame theory shows itself to be much more than a method for information retrieval or "heuristic search." One such answer is that Charniak's reference problem simply does not arise in the sense that the Birthday-Present-Frame-Keeper will already have assigned the new kite to a slot for "returned gifts" before the pronoun is processed!

The Demon Solution to the Reference Problem

Charniak's 1972 research on constructing a model for comprehending children's stories predates the articulation of the frame idea, but presages important notions of that concept in several ways. In particular, his formal realization of a *frame* was in the form of *base-knowledge* about a large variety of situations that arise in the context of these stories. The mechanism of his program was for the content of sentences to evoke this base knowledge with the following effect: demons were created to monitor the possible occurrence in later sentences of likely (but not inevitable) consequences of the given situation. Thus, for our story fragment the birthday

knowledge creates expectations about the need for participants of the party to buy presents and the possible consequence of having to return these gifts. Hence, these demons expect the possibility of Jack already possessing the present and the resulting need for Janet to return *it*, where *it* is known to be the present.

It is interesting to note that this demon about presents cannot be applied unless several common sense inferences are made. For example, the demon might be a formal representation of the following fact:

```
If a person P1 does not want a present X, then
he may request the giver P2 to return the present.
```

For the demon to be applied, the antecent, i.e. that P1 does not want the present X, must be triggered by an assertion in the database being created to represent the meaning of the story. (The database may be thought of as containing the kernel sentences of the story with each word mapped into its proper sense and reference problems disambiguated). But the story does not include explicitly all important facts. Look back at the story. Some readers will be surprised to note that the text itself does not state (a) that the presents bought by Penny and Janet were for *Jack*, (b) that the *top* bought by Janet was intended as a present, and (c) that having an object implies that one does not want another. All of the above facts are inserted into the database by other demons made activated by the birthday frame. Thus, Charniak's notion of a demon is particularly satisfying because not only do demons handle reference problems, they are also used to insert into the story database various reasons, motivations and purposes not explictly stated.

In the current view of frame theory, this insertion of unsaid information assumes a paramount role. From this point of view, it is not accidental that so much is left out of the story. Rather the typical situation in comprehension is to be faced with a set of clues that evoke a rich and detailed knowledge structure, the frame, that supplies the unstated details. Naturally, these defaults may be inappropriate for some situations and, in those cases, the text must supply the exceptions. But in general, reference, deduction and comphrehension are made possible only by knowing a great deal about the situation being discussed.

The Virtue of Particular Knowledge

AI is a young science and the ideas that we have discussed in previous sections are under constant development, elaboration and change. However, it has been our goal to give the reader a feeling for the kind of enterprise that AI is rather than a final statement on its theoretical form.

A central point has been the fashion in which many kinds of knowledge play a role in cognitive processes. In particular we considered for the reference problem the need for Real World versus Syntactic Knowledge and, more profoundly, Particular versus General Knowledge. A major point of our analysis, i.e. of the Knowledge as opposed to Power theory of intelligence, is that Particular Knowledge can often do the job for which General Knowledge is traditionally thought to be necessary.

It is encouraging that there are parallel issues in the theoretical writings of Piaget and the Geneva School of "Genetic Epistemologists". In particular Piaget's description of the knowledge of

the young ("pre-operational") child is much closer to the frame-specific knowledge paradigm than to the generalist power paradigm.

Self-Knowledge

In the next two chapters, we leave the context of Frame Theory in order to broaden our insight into the notion of AI as a new kind of Epistemology. To do this, we shall discuss two concerns which are central to AI theories but rarely appear in more traditional formulations of knowledge and intelligence: the first is the formal representation and use of self-knowledge and the second is the importance of control structures.

Kinds of Self-Knowledge

As we shall see, an interest in Self-Knowledge will not take us outside our concern for understanding the comprehension process. Indeed, we shall find that this kind of knowledge plays an obvious role in allowing the system (whether computer model or individual) to answer questions about its own performance as well as a role in the less obvious situation of handling special cases that arise in parsing the syntactic structure of the utterance.

There are many kinds of self-knowledge. Perhaps the simplest is the ability to observe one's own behavior, allowing the individual or program to answer elementary questions about its actions. A more skillful use of self-knowledge would involve being able to take account of these observations and direct one's problem solving accordingly. Still a more sophisticated program might, in certain circumstances, utilize this self-observation to *learn* by debugging faults in its procedures, generalizing its knowledge, or perhaps suitably specializing it to form more efficient techniques for particular problems. We shall illustrate each of these kinds of self-referential

abilities below in relation to the Blocks World, a favorite mini-world for AI programs in which a one-armed robot is asked to build various structures from blocks placed on a table (figure 1).

The SHRDLU Program and the Blocks World

Winograd's SHRDLU program represented a tour de force of design and an important benchmark in the realm of natural language comprehension by computer. In this section, we introduce the program briefly and in the next two illustrate several ways in which it manifests self-knowledge.

SHRDLU is a program that converted such sentences as "Pick up the big red block and put it in the box" into commands for a blocks world problem solver. The success of its comprehension was demonstrated by the program successively carrying out the instructions in a simulated world displayed on a cathode ray screen which contained a one-armed robot, a table, a box and a collection of cubes and pyramids.

The progam is capable of comprehending sentences that reached the degree of complexity illustrated by the following examples:

(1) Find the block which is taller than the one you are holding and put it into the box.

(2) Will you please stack up both of the red blocks and either a green cube or a

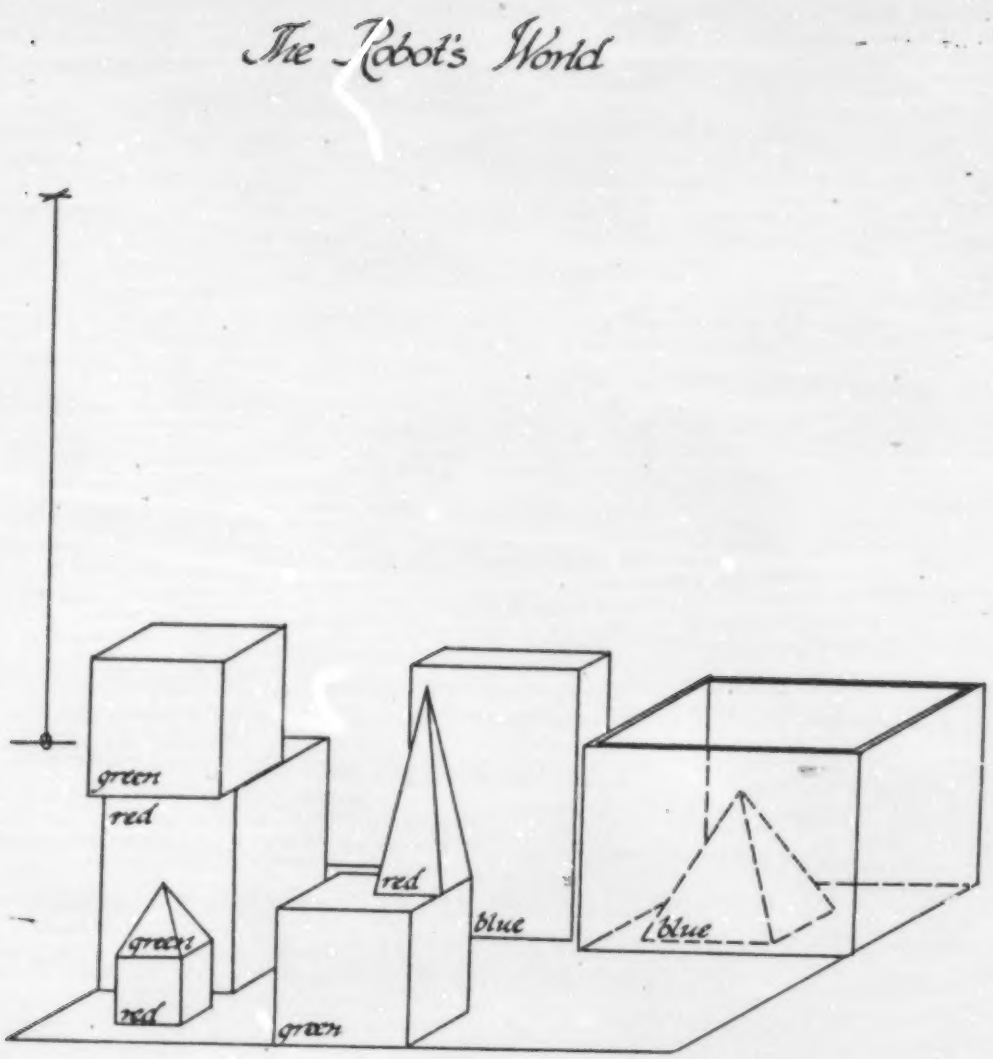


Figure 6
"Pick up the big red block."
From [Winograd 1972 p. 8].

pyramid.

(3) Does the shortest thing the tallest pyramid's support supports support anything green?

In SHRDLU, there are a multitude of different knowledge structures. For example, the system maintains event-history lists recording each event, why it occurred and what it accomplished; numerical coordinate data structure recording the dimension and location of objects; a symbolic database in which assertions about the state of the world, about global knowledge and about active procedures are stored; and a Fodor and Katz style marker hierarchy [Katz 1964] indicating the relation between words. But, perhaps most significantly, a major part of the systems knowledge was represented directly as procedures, thereby achieving far more flexibility and power than more traditional "declarative" representations. These different knowledge packets are used in many ways: to answer questions, to perform actions, to update its knowledge.

Answering Elementary Why Questions

An elementary example of self-knowledge is found in SHRDLU's capacity to generate names for assertions in its database, events, noun groups and other entities. With these names, the system can refer back to these entities, associating them with times, places and other information. This allows the system to answer elementary kinds of questions about its behavior. Figure 7 illustrates this ability.

23. had you touched any pyramid before you put the green one on the little cube?

YES, THE GREEN ONE

The system deduces from its memory what event is being referred to, and can relate other events to it in time. The analysis includes complex tenses of verb groups, and there are special facilities for keeping track of the various forms of irregular verbs like "have".

24. when did you pick it up?

WHILE I WAS STACKING UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE.

A time is described by naming the main goal which was being worked on.

25. why?

The system remembers immediately previous sentences to understand questions like this and the following ones.

TO GET RID OF IT.

By keeping track of selected parts of the original subgoal tree, the system has some understanding of its own motives.

26. why did you do that?

TO CLEAN OFF THE RED CUBE.

"That" refers to the action mentioned in the answer.

27. why did you clear off that cube?

Here, "that" means the one just referred to.

TO PUT IT ON A LARGE GREEN CUBE.

28. why did you do that?

TO STACK UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE.

29. why did you do that?

BECAUSE YOU ASKED ME TO.

It remembers what it did, not how the request was worded.

Figure 7

Examples of SHRDLU's ability to answer WHY questions. From [Winograd 1972, p. 13].

Self-reference in Grammatical Systems

A second level of self-referential ability in which the system can not only answer questions, but make decisions based on self-knowledge, is also illustrated by SHRDLU. This is the manner in which the system has access to not only the current state of the parse, but also to the past history of the parse. This is illustrated by the fashion in which Winograd's program parses conjunctions.

To parse sentences with words like "*and*", the system interrupts upon detecting the conjunction, notes the last constituent formed and attempts to find a duplicate constituent following the conjunction. If unsuccessful, it looks further back in the history of the parse until a substructure can be found that symetrically appears following the conjunction. A similar capability for self-reference is found in Woods Augmented Transition Network approach to parsing [Woods 1970]. Elements of this approach are discussed below in the section on Controlling Grammatical Computations.

It is interesting to note that "*and*" is not handled simply by augmenting the basic Systemic Grammar of SHRDLU. By some standards, this moving outside the formalism of one theoretical mechanism would be unaesthetic by being contrary to Occam's Razor. But experience has shown that the management of diverse knowledge packages is the rule rather than the exception in the architecture of complex problem solving systems. The obligation is simply one of clearly defining the interactions between these knowledge packets.

Debugging Knowledge

Recently, the Blocks World has served as a domain for investigating self-knowledge of a deeper kind. This has been in the form of a system called HACKER [Sussman 1974] which is capable of learning to solve block construction problems of the complexity that SHRDLU can handle. (Remember that for SHRDLU, the problem solving procedures are pre-programmed and not learned.) The learning is achieved through HACKER possessing an expertise in debugging. It is our belief that debugging knowledge is absolutely fundamental to the acquisition and operation of any program as complex as human thought and language.

Although this work was not done specifically as part of a natural language comprehension system, it provides the basic understanding to enable such a system to answer questions about why a particular method was used, in a far deeper fashion than SHRDLU's elementary abilities allowed. Knowledge of one's own problem solving strategies suuports an understanding that not only could discuss the chronology of events but could also discuss the underlying plans and their evolution through succeeding stages of generalization and debugging.

To illustrate this, we shall develop an example from Sussman. HACKER initially might be compared to a small child interested in constructing block structures, but not very knowledgeable about the problems involved. HACKER begins with the ability to place one block on top of another (by executing the primitive PUTON) and with an elementary planning capability. A typical elementary goal is to construct a tower from three blocks, expressed formally to the program by the conjunction (GOAL (AND (PUTON A B) (PUTON B C))). HACKER attempts to solve this problem in a "linear" fashion by first planning to achieve (PUTON A B), and then as an

independent second goal (PUTON B C). This produces the sequence of block states illustrated below:

```
                Trace of (ACHIEVE (AND (ON A B) (ON B C))

state 1:  A     B     C

                A
state 2:  B     C       ;PUTON A B has just been executed.

state 3:  A     B     C     ;The top of B has been cleared as a required
                ;pre-requisite of PUTON B C.  HACKER is one-handed
                ;and must lift blocks from the top.

                B
state 4:  A     C       ;PUTON B C has just been completed.
```

Obviously a bug has occurred. The action of clearing the top of B as a pre-requisite to achieving (ON B C) resulting in state 3 has undone the first sub-goal (ON A B) whose purpose was to place block A on block B.

One solution to this difficulty which AI programs of the past might have attempted (and which occasionally children pursue), is to return to the original state and try some different construction strategy, without attempting to learn anything from the current failure to guide the next attempt. HACKER contains a new kind of knowledge, namely knowledge of how to debug. This debugging knowledge knows that linear solutions are only first approximations and that sometimes interactions between the sub-goals of a conjunction must be taken into account. In this particular case, HACKER recognizes that the pre-requisite of one sub-goal CLEARTOP conflicts with the purpose of a brother goal. In such situations, it applies the debugging strategy of seeking

a reordering of the brother goals such that the goal with the pre-requisite is achieved first. In this particular problem, HACKER

(1) recognizes that the bug is due to the general bug-type of "pre-requisite clobbers brother goal".

(2) applies the debugging strategy of reordering the goals.

(3) edits in long-term memory the procedure for constructing towers of three blocks to take account of this ordering requirement.

(4) records a comment to itself for subsequent problems of this kind that states that when building block towers of any size, ON chains should be ordered from the bottom up.

Thus, HACKER exhibits both expertise about debugging and about repair, coupled with the ability to examine its own problem solving goals and actions so that it is able to apply this debugging expertise to its own reasoning. It represents a model of skill acquisition whose competence is based on self-knowledge about its behavior and meta-knowledge about the nature of plans and bugs.

We emphasize the importance of debugging skill because a vital element of intelligence is knowing how to handle a vast variety of situations. We cannot do this by knowing all about these situations because each real world situation is different. What we do need is:

(1) knowing how to tell which old situation is sufficiently like the present one, and

(2) knowing ways to adapt -- "debug" -- the old procedure for handling the old problem to a new procedure that can deal with the new situation.

Thus, subjects like analogy, similarity, metaphor are at the heart of the new formalisms. In the older methods of AI, linguistics, logic, and psychology, these were embarassing, hard-to-explain phenomena. In the new, debugging technology, they become concrete, manipulable types of knowledge.

Finally, without a theory of debugging, our image of frame-based comprehension would have a fatal flaw. Understanding by using large packets of previously known information reequires the ability to *debug* these knowledge structures for the particular application at hand, if the system is to be able to deal with novel situations.

Control Structures

We have seen the way in which AI epistemology is opening new areas of study in its concern for self-knowledge and, in particular, debugging skill. A second dimension along which AI's approach to epistemology is opening new horizons and fundamentally differs from older approaches is its attention to *control structure*. Indeed the idea of control structure could scarcely exist until one begins to see knowledge as made up of active agents which need to be controlled.

Controlling Grammatical Computations

"First-order" theories such as those composed of generative grammars are constructed under strong assumptions of modularity and independence. These assumptions are heuristic ones that enable the epistemological analysis to get underway, but include an expectation of a subsequent need to study the interactions. Controlling computation by choosing between the relevance of alternative chunks of knowledge, deciding when a given approach should be terminated, and generally guiding the reasoning process are among the important kinds of interactions which must be eventually understood.

When context-free grammars were first applied to the problem of comprehending text (rather than only generating grammatical sentences), it was immediately recognized that more structure was needed if the parsing times were to be kept within reasonable bounds. The essential problem is to provide information for choosing whether a given rule should be applied. It is this basic concern that characterizes the basic approach of both Augmented Phrase Structured Grammars [Heidorn

1975] and Augmented Transition Nets [Woods 1970]. The ATN approach has served as the basis of several important language programs and we shall discuss it further below. (See for example [Simmons 1973] in which ATNs are used both for parsing into an internal representation and for generation of English output from this representation.)

Augmented Transition Networks

The Augmented Transition Network formalism for expressing grammars developed by Woods was designed to extend the power of Finite State recognition grammars. His first step was to generalize the finite state graph to allow recursion, thus essentially obtaining the power of a context-free grammar while still preserving the perspicuity of a network representation. A recursive call is indicated by the label of an arc being a state name rather than a terminal. This directs the recognizer to jump to that state, run until acceptance, and then return control to the state pointed to by the arc. Essentially a pushdown store has been provided to keep track of the bookkeeping involved in allowing recursion. Figure 1 illustrates a Recursive Transition Net. An example of recursion occurs in the interpreter's transition from state S to state Q1 by means of the recognition of a noun phrase.

Woods' next step along the dimensions of adding control was to generalize these recursive transition networks further and provide for both a preference ordering on the arcs exiting a given node and arbitrary predicates which act as filters and must be satisfied for the node to be pursued. See figure 2 for a piece of such an Augmented Transition Net.

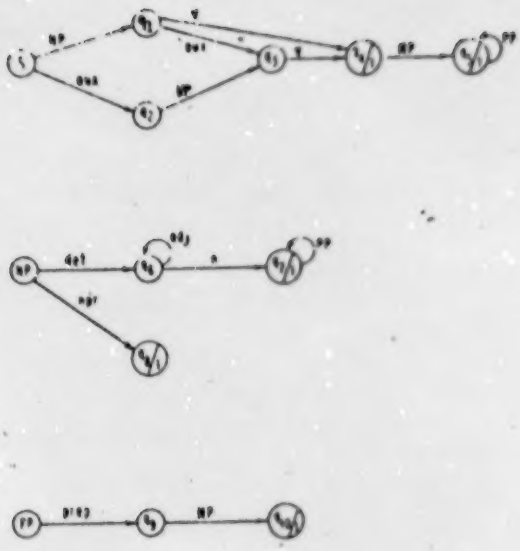


Figure 8

A Sample Recursive Transition Network
From [Woods 1970, p. 592].

BEST AVAILABLE COPY

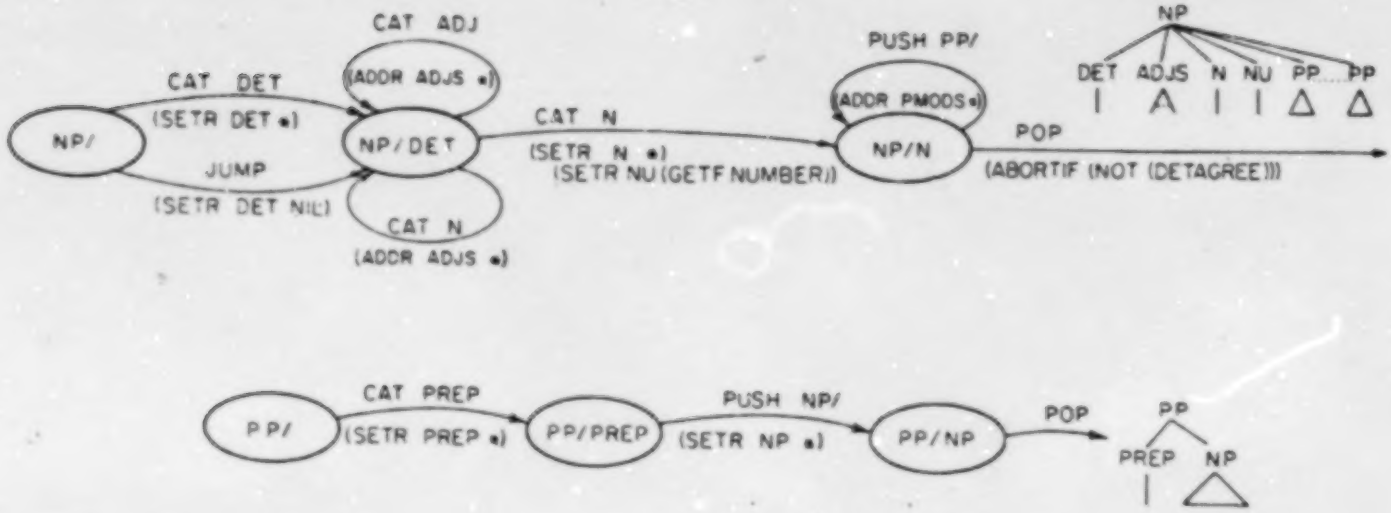


Figure 9

A Small Transition Network Grammar

From [Woods 1974, p. 137]

With the additional augmentation of allowing the system to maintain registers holding various pieces of information found during the parse, the ATN formalism has the full capabilities of a transformational grammar, but incorporates these abilities in a form that admits of far more efficient parsing algorithms than those previously proposed for transformation grammars.

> The addition of more powerful control information is only one of the arguments Woods advances for his ATN approach. Others relate to the perspicuity of such a representation, the efficiency involved in merging grammatical rules into common states, and the power obtained from the use of registers to allow context dependence and flexible structure-building activities.

Thus ATN's can be seen as basically addressed to the issue of control. The grammar is organized into states representing natural decision points. The possibilities exiting a state are ordered (as in production systems) and tests are placed on each arc to determine its admissability. The resulting formalism has proved to be successful and now serves as the basis of Woods Lunar system [Woods 1974] and several systems constructed by Simmons [1973; 1975].

Winograd's SHRDLU also addressed the control issue. Woods added control knowledge to a state-oriented description. Winograd added control knowledge by pursuing an entirely procedural approach. He represented grammatical knowledge as programs. These programs, being procedures, could have conditionals, loops, variables, or recursion. Figure 3 illustrates some simplified programs for recognizing sentences, noun phrases and verb phrases (expressed as flowcharts rather than in Winograd's Programmar formalism).

Woods preferred to preserve some element of declarative representation by clearly naming the states: Winograd preferred the freedom of an entirely procedural description. The approaches are computationally equivalent [Winograd 1972] and both represent the fashion in which first order grammatical theories can be extended to support efficient computation.

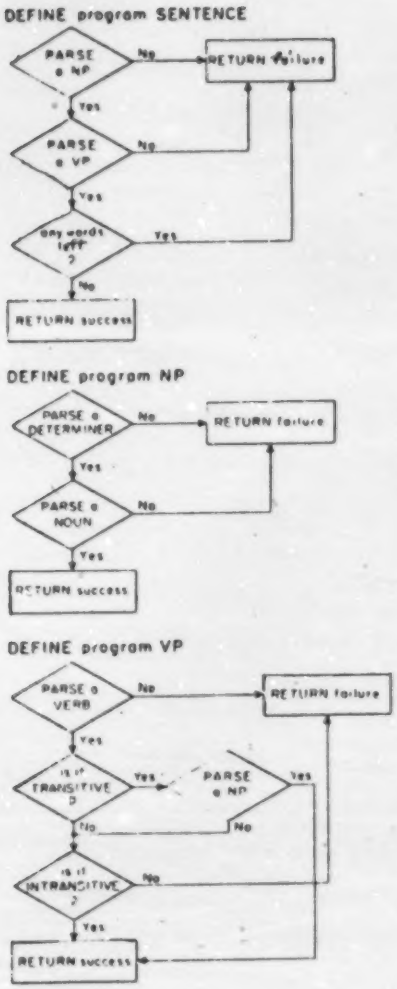


Figure 10

Flowcharts of simplified PROGRAMMAR grammars for noun phrases, verb phrases and sentences. From [Winograd 1972, p. 83].

BEST AVAILABLE COPY

> Of course, most AI theories are computationally equivalent just as most programming languages can through more or less labor be made to emulate each other. This is not, however, the real issue. THe critical question is the perspicuity of the language or representation for the particular applications envisioned. Woods emphasizes the virtues of a net description: Winograd prefers expression directly as programs.

An Issue for the Future -- Deterministic Parsing

The nature of these ideas is illustrated by going more deeply into a particular theoretical issue. A concern for control structure raises an obvious question for the future: will it be possible to organize grammatical knowledge (including interactions with other kinds of knowledge) in such a way that back-up is eliminated (without recourse to parallelism which is ultimately equivalent to back-up) at least for those sentences which one intuitively feels do not have a multitude of interpretations.

This goal of eliminating exhaustive search represents a natural concern of the Computational Epistemologist. First define the basic knowledge space. Then analyze its procedural structure. Viewing the process of finding a solution as a search branching at choice points, the question arises: "How is this branching controlled?" One possibility is to allow for backup. A second is to utilize parallelism. A third is to have diagnostic experts that can guide the decision at choice points so that backup is minimized or perhaps even eliminated.

> This evolution has occurred most notably in the development of programs capable of symbolic integration. An early program by [Slagle 1963] had knowledge of a basic set of integration formulas but combined this knowledge by means of a search. Recent programs following Moses' design of SIN [Moses 1967] have a deeper understanding of the nature of integration and do not require any search to find the integral. We do not necessarily expect language

> 1967] have a deeper understanding of the nature of integration and do not require any search to find the integral. We do not necessarily expect language programs to evolve to the extreme form of deterministic problem solving that the integration programs have reached, but we do expect a similar kind of evolution in the same direction.

Insight into grammatical knowledge and knowledge representation techniques has reached the point where one can seriously consider a backup-free approach. The issue is not decided, but the suggestions are promising. For example, a backup-free scheme has recently been proposed by Marcus [1975]. He calls his approach "wait and see parsing". The basic idea is that backup is made unnecessary if the system is provided with a limited look ahead of only a few words (or phrases) and if sufficient constraint is provided by semantic and pragmatic knowledge. He is currently constructing a case-frame based parser following these guidelines.

Another proposal by Riesbeck [1975] involves a somewhat more radical approach. Riesbeck thinks that grammatical knowledge should be thoroughly integrated with semantic and pragmatic knowledge to such an extent that it is no longer reasonable to speak of separate components. This represents the extreme position along a natural line of evolution of more and more intimate interaction between grammatical and other kinds of knowledge. Riesbeck has constructed a comprehension system that reflects this philosophy. Words in the sentence trigger "productions" (called demons by Riesbeck) whose goal is to build a representation of the sentence's meaning (using Schank's conceptual dependency structures [Schank 1973]). Some of these productions reflect grammatical knowledge such as generating the expectation for a noun following the appearance of an article. Others represent semantic and pragmatic information such as expecting a human recipient following an appearance of the word "give". Riesbeck proposes that such an organization has both conceptual elegance as well as the ability to parse deterministically.

A new "conceptual" grammar by Martin [1975] also promises to generate perspicuous, backup-free conversions from sentences into expressions in his semantic representation. Martin's work suggests the possibility that an internal representation might be closer to the "surface" structure of natural language than most workers have believed feasible.

Clearly there is an important interaction between grammatical knowledge and other kinds of knowledge. A purely pipeline model in which a grammatical analysis is followed by a semantic analysis and finally a pragmatic analysis is untenable. However, it is not yet clear which alternative among the following is preferable: (1) structuring the grammatical knowledge as a modular expert engaged in a dialog with other experts or (2) distributing the grammatical knowledge with other kinds of knowledge in such a fashion that it is no longer appropriate to talk about separate syntactic, semantic and pragmatic components. In any case, it is our expectation that in the next few years language systems will be constructed in which backup does not generally occur (except in the case of garden path sentences which are equally confusing to people). This is based on the observation that most procedural theories initially require backup, not because of the intrinsic properties of the problem area, but rather only because sufficient knowledge to disambiguate alternatives has not yet been incorporated into the system. This knowledge has gradually been formalized (as our earlier discussion of work by Winograd and Woods showed). Consequently, we are evolving towards the stage where knowledge is available to make the right choices at each decision point.

The Classical, Romantic and Modern Periods of AI

A Change in Perspective

In the preceding chapters, we have tried to give the flavor of the new perspective from which AI approaches traditional problems in linguistics and epistemology. Specific forms of this general change in perspective are related to a number of theoretical issues:

- the relative importance of procedural as opposed, usually, to "declarative" or "propositional" knowledge;

- the shift away from traditional logistical concepts (axiomitizations, logical deduction) for the representation and manipulation of knowledge to more highly structured and particular representations;

- liberation from the need for highly uniform transformational methods by allowing procedural sentries or "frame-keepers" to populate the knowledge system.

In each case we note a trend away from a traditional form towards a newer form. In each case experience has already shown us that such trends can go too far, and that there is often good reason for using the traditional forms. So, looking back over the brief history of AI, we see a dialectic process with three phases which might be termed the Classical, Romantic and Modern Periods of AI.

In the earliest work (which might be termed the *Classical Period* extending from the mid-fifties through the early sixties), knowledge is represented in very traditional ways: for example as subject - predicate propositions or as important but relatively non-fundamental extensions of this to property lists. Reasoning is accomplished through uniform techniques such as search or theorem

proving.

Typical programs of the Classical Period were often criticized by laymen on the grounds that "people don't think like that." (Of course, this reaction may be totally wrong, since people do not necessarily know what deep processes lie under the conscous layers of thinking.) However, whether or not these objections are right in principle, the programs of what we shall term the Romantic Period (extending from the late sixties to early seventies) do not evoke them nearly to the same extent. On the contrary, they sometimes appear to be nothing more than a new language for introspective material. Striking examples are the Evans ANALOGY program [Evans 1968] and the Winston learning program [Winston 1970].

This stage might be justifiably termed the Romantic Period for two reasons. One is the extent to which it represented an introspective endeavor while the second is a notable tendency to argue for first one approach and then another as a complete theory of intelligence. The period began with an emphasis on network-like systems [Quillian 1967] and ended with a concentration on the representation of knowledge in an entirely procedural form [Hewitt 1972, Winograd 1972].

Finally, in this the *Modern Period*, we begin to investigate more rigorously the conditions under which different kinds of knowledge are appropriate. Frames are extended from an essentially declarative data type to a procedural system through the incorporation of demons. Procedures, in order to allow debugging, are embedded in a web of declarative commentary specifying the purposes, requirements, bugs and effects of programs. Procedural knowledge thus serves as a basis for action while propositional knowledge provides a basis for understanding what one is doing and especially for knowing how to debug or change it.

Some Unanswered Technical Issues in Frame Theory

In this section, we present a sampling of the questions that make the investigation of the ideas expressed in this essay into a scientific enterprise. The frame approach to representing knowledge is suffiently novel that many of these questions are current issues of dispute in the community. We shall make no attempt to resolve them here. Our purpose is only to illustrate to the reader that AI is more than just a philosophic activity and that the use of computers requires that technical issues be resolved (and indeed serve to test the success of a given approach).

Learning:

A theory that makes extensive use of pre-stored knowledge obviously faces a critical question: how is such knowledge obtained? Or, in other words, how does such a theory allow novel situations to be comprehended?

Reference and Reasoning:

Given that the system knows that feathers can be used to tickle (i.e. could successfully answer a question to that effect), it still remains to describe how indirect reference or reference through description is accomplished. Presumably, there are some tickling instruments that we have never before considered, but that we nevertheless are able to understand in that role. How are such inferences made?

Difficulties with Sentries:

There are a variety of technical problems that arise in implementing the idea of a sentry as a demon. These include handling conflicts (i.e. situations where more than one demon is triggered

by the current situation and each has an alternative meaning), forgetting (i.e. when should a demon become inactive), and context-switching (i.e. how are leaving and returning to contexts handled).

Representation:

In many ways, the frame idea has served as a metaphor for us to present various ideas about the representation of knowledge -- both declarative and procedural. However, there are many representation questions that must be resolved in formalizing the notion. For example, how homogeneous is our representation of language to be? Are sentries (demons) to be represented in the same form as the basic frames? Are current computer languages such as Lisp adequate or do we require new programming constructs?

Control:

Is knowledge of control implicit or explicit in a frame system? Control is explicit when in addition to basic facts the system also includes knowledge about when to apply these facts and how to resolve conflicts. Control thus becomes just another kind of knowledge to the system. On the other hand, control is implicit through the representation of knowledge as procedures, with ordering in the program, conditionals and calls determining the sequence of application. Which approach is to be preferred?

Level Interactions:

We have posited the existence of frame-based knowledge for verbs, nouns, contexts, and themes. Technical questions arise about how these various levels of knowledge interact. Is the organization essentially *hierarchical*, with each level looking only at the results of the interpretation

of the level below it. Or do all levels examine the text (perhaps in addition to examining each other)? Is communication occasionally *heterarchical* with higher levels sometimes directing the analysis of lower levels and at other times being in turn directed by the results of the lower level analysis?

This chapter represents the conclusion of our theoretical discussion of AI as a new style of epistemology and its impact on the understanding of language and comprehension. In the next chapter, we consider the relevance of these ideas to education.

The Impact of Artificial Intelligence on Education

If the Machines Get Smarter, Will the Children Get Smarter?

The relevance of Artificial Intelligence to Education can be discussed along two different dimensions corresponding to the narrow and broader views of the subject as "machine intelligence" and as "epistemology". The former view evokes the prospect of more useful machines, sometimes described under the label "Intelligent Computer-Aided Instruction" [Brown 1975]. The other view has possible consequences of a much deeper sort, for it can be taken as projecting a total restructuring of the content and the theory of education. But even if this radical prediction is not fulfilled, the cognitive style of Artificial Intelligence has very immediate and realistic suggestions about what children should be taught and how.

> The two views can be stated in a more dramatic way. The organizers of the HUMRO-NSF conference on "Ten Year Predictions of the Impact of Computers on Education" [HUMRO 75] chose as the theme of one session: "If the machines get smarter, will the children get smarter?".
>
> We think the deeper question is this: "Suppose that *smart* machines can be constructed. Should educators look to the machines themselves as instruments to make kids smart, or should they rather look at the principles which led to the machines becoming more intelligent to see whether these principles could be applied to children?" There is, of course, no need to choose. Our intention in so posing the question is to draw attention to the richer, but almost entirely neglected, second view.

This paper is not the place for a full discussion of the relevance of Artificial Intelligence to Education. Instead, we shall develop briefly three representative issues:

(1) the first is the impact on education of the development of computer-based

tutors with reasonable language comprehension abilities. We shall develop this topic under the heading "Intelligent Computer-Aided Instruction."

(2) the second is the deeper question of whether the content of AI theories of language can have an impact on the very substance of linguistic studies in the schools, not just in the manner in which the current material is taught. We have chosen to call this topic "The White Box Approach to Language Studies".

(3) the third is whether the basic goal of AI in seeking unified theories of knowledge can be expected to have a profound impact on the entire educational curriculum, with the effect of decreasing the currently accepted division of education into disparate subject areas. We shall discuss this topic under the title "The Articulate Learner".

Again we remind the reader that our discussion is intended only to give the flavor of a few kinds of interaction that Artificial Intelligence and Education might have. For a discussion of related areas of research in psychology, linguistics and AI, see [Winograd 75b]. For a deeper discussion of the particular interactions discussed here, see [Minsky-Papert 1974], [Papert 1972], [Goldstein 1974])

Intelligent Computer-Aided Instruction

The question cited earlier "If the machines get smarter, will the children get smarter?"

immediately evokes the comment "certainly not automatically". This follows from the deeper observation that the question is not primarily about machines, but rather about the nature of human intelligence and the effectiveness of teaching. To see this, transpose it into the two sub-questions:

(1) "Could an idealized personal human tutor make a kid smarter?" and

(2) "Could all or some of the functions performed by this tutor be mechanized?".

Answers to the first question will vary depending on what one thinks about controversial issues in the theory of (human) intelligence. But there would certainly be less variation in answers to the question: "Could a highly intelligent, empathetic personal tutor enhance the intellectual development of children?" (Whether this enhancement is to be described as increasing the child's intelligence or the effectiveness of his use of it is not germane here.)

There are very few programs that can lay claim to fulfilling the functions of an intelligent, insightful tutor. However, important ground-breaking research has been done by Carbonell [1970] and Collins [1975a] working on the SCHOLAR Project, and by Brown [1974] in his development of the SOPHIE program. The SCHOLAR Project has developed a series of successively more sophisticated computer-based tutors for such domains as geography. The project has involved the development of natural language comprehension modules, the construction of domain-oriented problem solvers based on a semantic net approach to representation, and, most notably, the procedural representation of tutorial strategies.

Recent work by Collins [1975b] has involved a deep interaction between psychology and Artificial Intelligence in the form of studying the kinds of reasoning that people typically use to handle questions in *open domains*, i.e. concerning topics about which they know a great deal but not all of the relevant facts. People are typically quite good at making plausible inferences in such contexts. Collins' goal is to represent such reasoning strategies in a procedural way and then make that knowledge available to the SCHOLAR tutor for the purpose of allowing the tutor to help a student debug his own reasoning style.

Brown [1974] has developed a computer-based laboratory in which a student can learn electronic troubleshooting. The program SOPHIE contains an electronics problem solver that embodies a genuine understanding of the circuits involved, a natural language component that allows English question-answering to occur between the student and program, and an understanding of certain kinds of tutoring strategies. We shall not develop the details of Brown's program further, except to remark on the following points:

(1) the program is a milestone in C.A.I. simply because of the extent to which it understands its subject matter. Previous C.A.I. programs (and indeed most current ones) are notorious for their ignorance. The slightest deviation from the expected response on the part of the student in his answer results in the program erroneously judging the answer as incorrect. Brown's program utilizes a variety of AI techniques to simulate the circuits under study and accurately predict the consequences of various faults. This kind of domain competence is essential as a foundation for any C.A.I. tutor, and we expect the incorporation of AI problem solvers into C.A.I. programs to become more and more common in the coming years.

(2) the tutoring strategies of SOPHIE are concerned with the nature of reasoning involved in building up a model of a problem from incomplete evidence. This is a fundamental kind of human reasoning. It addresses the basic issue of what kinds of questions a student should ask in attempting to understand a new subject, device or electronic circuit. SOPHIE is able to judge whether a measurement requested by the student (as part of the process of tracking down a fault in the circuit) is a reasonable question on the basis of what the student already knows. If not, i.e. if the answer to the question is already entailed by information previously obtained by the student, then the program draws the student's attention to the redundancy and discusses the information which the student knew but failed to consider.

This study of human cognitive processes through the formalisms developed by Artificial Intelligence and the incorporation of that understanding into C.A.I. programs is only in the formative stages. But it is our hope that in the coming years, the concept of personal computer-based tutors available to any student at any time will become more and more of a reality.

The White Box Approach to Language Studies

In the previous section, we saw the application of Artificial Intelligence research on language comprehension and problem solving to the design of Intelligent C.A.I. programs. In this section, we develop a more radical application.

The I.C.A.I. tutors, no matter how intelligent, are essentially *black boxes* from the student's

perspective. The possibility exists that the concepts developed to design those programs might themselves be worthy objects of study by the students. We propose calling this a *white box* approach. Unlike the "black box" approach, the point here is precisely to encourage the young students to look inside the box, to ask how it works and to gain insight into their own language by so doing.

An example of the white box approach is illustrated by suggesting a new kind of *Language Laboratory*. We envision a computer-based environment in which students would have the opportunity to explore the structure of language by designing, programming and testing various kinds of language comprehension programs.

The virtues of such a laboratory approach lie in the active nature of the student's involvement, the possibility of the student developing his ideas in a personal way not rigidly limited by the teacher's approach (but reasonably circumscribed by the goals of designing a successful program), and the exposure to new computational theories of language that are in many ways superior to traditional grammars as conceptual frameworks for the study of language at school.

As an example of the power of these new approaches to language, we mention the possibility that they hold out for a unified approach to the study of knowledge. This was illustrated earlier by our discussion of Frame Theory in which the same basic knowledge-representation techniques were seen to apply to grammatical knowledge as well as other kinds of knowledge concerned with themes, contexts and semantic meanings. The exciting possibility exists of diminishing the compartmentalization of school knowledge. Linguistic knowledge becomes less of a "different"

thing, apart from other knowledge.

Our series of roles for language understanding machines has moved progressively away from a focus on the machine to a focus on the linguistic ideas which make the machine possible. We conclude this essay by generalizing this approach beyond the linguistic domain.

The Articulate Learner

In this section, we develop further the notion of a unified approach to educational curricula by introducing the notion of *Artticulate Learning*. Our discussion is motivated by the question of why students who find language studies in school challenging and who do well in them often find the mathematics curriculum a confusing and impossible maze. A traditional answer is that there is such a thing as quantitative mathematical abilities and qualitative language-oriented abilities. Such a belief is mirrored in all of the Achievement Tests that students are asked to take in which there are clearly demarcated language and mathematics sections.

We should like to raise the possibility that the traditional mathematics curriculum of grades 1-12 lacks certain dimensions which make it understandably obscure compared to language studies. In particular, its basic failing is in the extreme difficulty which it presents for a student or teacher who attempts to be articulate about the problem solving process.

In language studies, a teacher can discuss the many dimensions of a story written by a student: its grammar, its theme, its narrative structure, the plot, the setting. In contrast, what can a teacher

discuss about the failure of a student to solve an arithmetic problem or an integral? For example, we have seen a child go to the blackboard and solve an arithmetic problem in the following way:

```
  35
+ 35
----
 610
```

The teacher involved had little to say besides observing that the answer was wrong. We believe that the earlier discussion of AI and Epistemology has some very definite suggestions for altering the mathematics curricula to allow the student and the teacher to be far more articulate about the knowledge and problem solving involved.

For the above example, we imagine an environment in which the first kind of knowledge that a student is exposed to is the structure of programs and the related ideas of planning and debugging. Skill in designing procedures lies at the heart of achieving competence in mathematics (and in writing essays as well). Given access to a vocabulary for programs, plans and bugs, we believe that student and teacher could be articulate about describing the particular algorithm used by the student in reaching the "610" answer, in identifying the bugs, and in debugging the addition program to yield a correct result. For the above example, we imagine a discussion similar to the earlier description of the HACKER program's reasoning regarding its problem solving in the Blocks World. The student had mastered an addition procedure for single digits. His goal in this problem is to apply that procedure as a *sub-routine* to the more complex goal of adding multi-digit numbers. His solution indicates that he followed the reasonable *linear plan* of assuming that the sum of multi-digit numbers could be achieved by adding the columns *independently*. Linear planning commits the problem solver to have some method of putting together the solutions of the

sub-problems which were arrived at independently. The answer of "610" indicates that our student utilized the simple artifice of concatenating the two partial solutions. The *bug*, of course, was in assuming the complete independence of the column additions. The *debugging* is accomplished by taking account of this interaction through ordering the columnar additions right-to-left and utilizing a "carry" *data structure*.

The reader may find a vocabulary involving such terms as linearity, bug, debugging, independence and data structure confusing; and must certainly believe that such a vocabulary is hardly useable with an elementary school student. This is true, of course, given the current curriculum.

However, our proposal is not simply to add this vocabulary to the traditional curriculum. It is more far reaching. The proposal is to redesign the curriculum to allow introduction of this and related concepts at an early stage.

This is being done in a preliminary way at M.I.T.'s LOGO project [Papert 1972] in which new computation-based approaches to mathematics, science and music are currently being developed. As an example, elementary school students are introduced to the above set of programming concepts and to traditional mathematical concepts in the context of programming a computer display to draw pictures of their choice (figure 1).

The children construct the programs themselves in LOGO, a programming language expressly designed for students. A successful LOGO experience involves not only the student being self-motivated to construct such programs (because of the enjoyment of creating pictures),

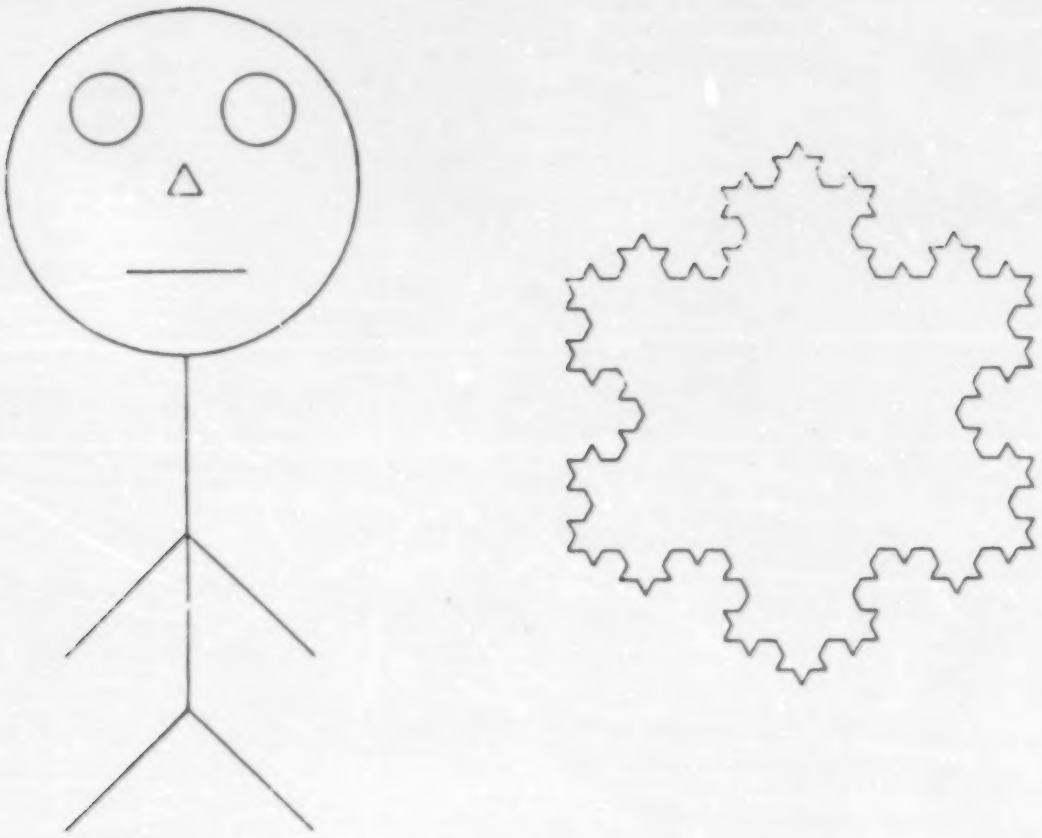


Figure 11

Typical Pictures Drawn by Children

From [Goldstein 1974, p. 9 & 21].

but also involves discussions between student and teacher, between students and student and between the student and himself in which the student becomes verbal about his reasoning and learns to use a vocabulary describing his plans, his programs, and, most importantly, his bugs.

The design of new educational environments in which the student is introduced to the concepts that AI has found essential to the construction of intelligent programs, is only in a beginning state. But our experience with children in teaching them Turtle Geometry, Computer-based Physics, and Computer-based Composing leads us to believe that the approach is promising and potentially revolutionary. Furthermore, it seems to lead to a situation where students can be articulate about their own problem solving, regardless of whether it is in the mathematical or linguistic domain.

Epistemology and Education

The role of a knowledge-oriented AI approach to the study of language comprehension has its counterparts in all areas of cognitive science, and each of these carries implications for education.

We have argued again and again that a major thrust of AI research has been to find representations of knowledge which lend themselves to effective integration into larger knowledge systems. The analogous problem for education is obvious: Are the representations of knowledge which have proven themselves effective in augmenting machine intelligence also of value in augmenting human intelligence? And, if not, does a similar inquiry into the forms of knowledge

suitable for human thinking yield a different set of modes of representation? Futhermore, the epistemological paradigm for the study of learning asks questions like: What knowledge does a good learner use in a given situation? What knowledge would make learning a simpler process than it appears to an observer who does not know this knowledge is being used, or to a learner who lacks it?

Very little attention has been paid to problems of this sort. But signs of a rich area of the utmost importance seem to be present in the results of such preliminary investigations as the construction of I.C.A.I. programs containing procedural models of human inference and learning, the design of new approaches to learning based on Debugging Theory, and the design of new curricula such as Turtle Geometry based on a notion of the Articulate Learner.

It is our belief that future explorations into the nature of language, of various subject domains and of the elements of intelligence itself -- if pursued from an epistemological perspective grounded in the computational metaphor -- will provide the foundation for deeper theories of the nature of language and of the nature of learning. Furthermore, we expect such theories to have practical consequences in the form of programs that comprehend language and in the design of new kinds of computer-based educational environments.

Epilogue

Controversies about Educability and Innateness

The general issues about the theory of education discussed in the last chapter cut across a number of acute current controversies about the relations between linguistic abilities in the developing child and other cognitive functions including learning. Different approaches to linguistics and to cognitive theory appear to lead theorists to very different opinions on questions such as the *domain-specificity* of language skills; their relation to particular brain structures; and the extent of their *innateness*. For example, one sees a very sharp opposition between the Chomskian and Piagetian lines of thinking [Chomsky 1975]. For the former, there is a definite *linguistic competence* which is extremely specific to language (i.e. has very little overlap with other cognitive abilities) and which is determined by innate biological structures. Adherents of the Piagetian school, on the other hand, tend to see linguistic competence as sharing major cognitive processes with other intellectual domains and to be explicable more in terms of a psycho-developmental process.

> Any such brief summary of complex theories necessarily over-simplifies the respective positions. Nevertheless, the oppositions, though more modulated, are real and of fundamental and urgent importance.

We see work in Artificial Intelligence as having a very direct bearing on this issue. This does not mean that we are able to *resolve* the issues. Rather we believe that *computational models* provide a theoretical overview necessary even for a clear statement of the issues. This belief is reinforced by the increasingly rapid spread of computational ideas into the work of psychologiests and even philosophers whose views fall on both sides of these controversies.

> See [Fodor 1975] for an examples of an extreme "nativist" position formulated within a computational theoretical frame by a psychologist who disagrees fundamentally with much of the "main-line" AI trends. See [Norman

1975, Bobrow 1975] for views of psychologists sympathetic to the kind of AI theory treated in this paper.

In conclusion, though the debates about intelligence and cognition rage both in and out of AI, and the reader may have objected to many of the specific observations that we have made regarding frame theory and other issues, we believe that there already has been this fundamental and profound change: that the language of computation has become the proper dialect for discussing the basic issues of both psychology and education.

Bibliography

[Brown 1974]
Brown, J.S., R.R. Burton and A.G. Bell: SOPHIE: a Sophisticated Instructional Environment for Teaching Electronic Troubleshooting, BBN Report No. 2790, Bolt Beranek and Newman, Cambridge, Ma., 1974.

[Brown 1975]
Brown, J.S.: Proposal for Knowledge-based Intelligent Computer Assisted Instruction BBN PROPOSAL NO. P75-CSC-17A, May 28, 1975. Bolt, Beranek, and Newman, Cambridge, Ma., 1975.

[Carbonell 1970]
Carbonell, H.: AI in CAI: an Artificial Intelligence Approach to Computer-Assisted Instruction, *IEEE Trans. Man-Machine Systems*, Vol. MMS-11, pp. 190-202, Dec. 1970.

[Celce-Murcia 1972]
Celce-Murcia, M.: English Comparatives, PhD. Thesis, Dept. of Linguistics, UCLA, Los Angeles, 1972.

[Chafe 1970]
Chafe, W.L.: *Meaning and the Structure of Language*, Chicago University Press, Chicago, 1970.

[Charniak 1972]
Charniak, E.: Toward a Model of Children's Story Comprehension, AITR-266, Artificial Intelligence Laboratory, MIT, Cambridge, Ma., 1972.

[Chomsky 1957]
Chomsky, N. *Syntactic Structures*, The Hague, The Netherlands: Uitgverij Mouton.

[Collins 1975a]
Collins, A., E.H. warnock, and J.J. Passafiume: Analysis and Synthesis of Tutorial Dialogues, in G. Bower (ed.), The Psychology of Learning and Motivation, Vol. 9, Academic Press, New York, 1975.

[Collins 1975b]
Collins, A., E.H. warnock, N. Aiello, and M. Miller: Reasoning from Incomplete Knowledge, in D. Bobrow and A. Collins (eds.), *Representation and Understanding*, Academic Press, New York, 1975.

[Drake 1975]
Drake, C.: Semantic Case Frames in the Understanding of Connected Discourse, Unpublished Masters dissertation, M.I.T., Cambridge, Ma., 1975.

[Evans 1967]
Evans, T.: A Program for the Solution of Geometric-Analogy Intelligence Test Questions, in M. Minsky (ed.), *Semantic Information Processing*, The MIT Press, Cambridge, Ma., 1968.

[Fillmore 1968]
Fillmore, C.: The Case for Case, in E. Bach and R.T. Harms (eds.), *Universals in Linguistic Theory*, Holt, Rinehard, & Winston, New York. 1968.

[Fodor 1975]
Fodor, J. A.: *The Language of Thought*, Crowell, New York, 1975.

[Goldstein 1974]
Goldstein, I.P.: Understanding Simple Picture Programs, AITR-294, Artificial Intelligence Laboratory, MIT, Cambridge, Ma., 1974.

[Green 1963]
Green, B.F., A.K. Wolf, C. Chomsky, and K. Laughery: Baseball: an Automatic Question Answerer, in E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*, McGraw-Hill, New York, 1963.

[Heidorn 1975]
Heidorn, G.: Augmented Phrase Structure Grammar, in R. Schank and B.L. Nash-Webber (eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, Ma., 1975.

[Hewitt 1972]
Hewitt, C.: Description and THeoretical Analysis (Using Schemata) of Planner, AITR-258, Artificial Intelligence Laboratory, MIT, Cambridge, Ma., 1972.

[HUMRO 1975]
Human Resources Research Organization: Ten Year Forecast for Computers and Communications: Implications for Education, Arlie House, Warrenton, Va., Sept. 1975.

[Katz 1964]
Katz, J.J. and J.A. Fodor: The Structure of a Semantic Theory, in J.A. Fodor and J.J. Katz (eds.), *The Structure of Language*, Prentice-Hall, Englewood Cliffs, 1964.

[Lindsay 1963]
Lindsay, R.K.: Inferential Memory as the Basis of Machines which Understand Natural Language, in E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*, McGraw-Hill, New York, 1963.

[Marcus 1975]
Marcus, M.: Diagnosis as a Notion of Grammar, in R. Schank and B.L. Nash-Webber (eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, Ma., 1975.

[Martin 1975]
Martin, W.: Conceptual Grammar, Internal Memo 20, Automatic Programming Group, Project MAC, M.I.T., Cambridge, Ma., 1975.

[Minsky-Papert 1974]
Minsky, M. and S. Papert: *Artificial Intelligence*, Condon Lectures, Oregon State System of Higher Education, Eugene, Oregon, 1974.

[Minsky 1975]
Minsky, M.: "A Framework for Representing Knowledge" in P. Winston (ed.), *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[Moses 1967]
Moses, J.: *Symbolic Integration*, MAC-TR-47, Project MAC, M.I.T., Cambridge, Ma., 1967.

[Newell 1973]
Newell, A.: Artificial Intelligence and the Concept of Mind, in R. C. Schank and K. M. Colby (eds.), *Computer Models of Thought and Language*, W.H. Freeman, San Francisco, 1973.

[Norman 1975]
Norman, D.A. and D. E. Rumelhart and the LNR Research Group: *Explorations in Cognition*, W. H. Freeman, San Francisco, 1975.

[Papert 1972]
Papert, S.: Teaching Children Thinking, Programmed Learning and Educational Technology, Vol. 9, No. 5, Sept. 1972.

[Quillian 1968]
Quillian, M.R.: Semantic Memory, in M. Minsky (ed.), *Semantic Information Processing*, The MIT Press, Cambridge, Ma., 1968.

[Riesbeck 1974]
Riesbeck, C.K.: Computational Understanding: Analysis of Sentences and Context, Working Paper, Fondazione Dalle Molle per gli studi linguistici e di comunicazione internazionale, Castagnola, Switzerland, 1974.

[Riesbeck 1975]
Riesbeck, C.K.: Computational Understanding, in R. Schank and B.L. Nash-Webber (eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, Ma., 1975.

[Rumelhart 1975]
Rumelhart, D.E.: Notes on a Schema for Stories, in D. Bobrow and A. Collins (eds.), *Representation and Understanding*, Academic Press, New York, 1975.

[Schank 1972]
Schank, R., N. Goldman, C. Rieger, and C. Riesbeck: Primitive Concepts Underlying Verbs of Thought, AIM-162, Stanford University, 1972.

[Schank 1973]
Schank, R.C.: Identification of Conceptualizations Underlying Natural Language, in R. C. Schank and K. M. Colby (eds.), *Computer Models of Thought and Language*, W.H. Freeman, San Francisco, 1973.

[Schank 1975]
Schank, R.C.: Using Knowledge to Understand, in R. Schank and B.L. Nash-Webber (eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, Ma., 1975.

[Simmons 1973]
Simmons, R.F.: Semantic Networks: Their Computation and Use for Understanding English Sentences, in R. C. Schank and K. M. Colby (eds.), *Computer Models of Thought and Language*, W.H. Freeman, San Francisco, 1973.

[Simmons 1975]
Simmons, R.F.: The Clown's World, in R. Schank and B.L. Nash-Webber (eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, Ma., 1975.

[Slagle 1963]
Slagle, J. R.: A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus, in E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*, McGraw-Hill, New York, 1963.

[Sussman 1975]
Sussman, G.J.: *A Computational Model of Skill Acquisition*, American Elsevier, New York, 1975.

[Wilks 1975]
Wilks, Y.: Natural Language Understanding Systems within the A. I. Paradigm: A Survey and Some Comparisons, Unpublished paper prepared for the National Institute of Education.

[Winograd 1972]
Winograd, T. *Understanding Natural Language*. New York: Academic Press, 1972.

[Winograd 1975a]
Winograd, T. Frame Representations and the Declarative/Procedural Controversy, in D. Bobrow and A. Collins (eds.), *Representation and Understanding*, Academic Press, New York, 1975.

[Winograd 1975b]
Winograd, T. Artificial Intelligence and Language Comprehension, Unpublished paper prepared for the National Institute of Education.

[Winston 1970]
Winston, P.: Learning Structural Descriptions by Examples, reprinted in Winston, P. (ed), *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[Woods 1970]
Woods, W.: Transition Network Grammars for Natural Language Analysis, *Communications of the ACM*, Volume 13, Number 10, October, 1970.

[Woods 1974]
Woods, W.: *Natural Language Communication with Computers* BBN Report No. 2976, Volume I, Bolt, Beranek and Newman, Cambridge, Ma., 1974.